Pengamanan Data dengan Menggunakan Teknik Kriptografi Public RSA dan Knapsack

Rohani Sufi Yanti Simbolon[1], Parasian Silitonga[2]
[1,2,] *Fakultas Ilmu Komputer Universitas Katolik Santo Thomas Medan, Indonesia*

ARTICLE INFORMATION

Received: April, 18, 2021
Revised: April 22, 2021
Available online: April,28,2021

KEYWORDS

Sistem keamanan data, algoritma kriptografi asimetri RSA dan Knapsack

CORRESPONDENCE

E-mail: rohanisimbolon55@gmail.com[1]
parasianirene@gmail.com[2]

A B S T R A C T

Data security and confidentiality issues are very important in an organization and in a person. Especially if the data is in a computer network that is connected or connected to a public network, for example the internet. Of course, this very important data was seen or hijacked by unauthorized people. Because the delivery security system is very broad in scope, this section is limited by using the RSA and Knapsack asymmetric cryptographic algorithms which include encryption and decryption processes. This data security system using the RSA and Knapsack asymmetric cryptographic algorithms can solve problems that often occur, such as data theft, data corruption and misuse, and users can maintain the confidentiality of their important data / files.

PENDAHULUAN

Pada masa sekarang, komputer dan aplikasi telah berkembang pesat penggunaannya, sehingga dapat dikatakan keberadaannya sangat melekat pada kehidupan sehari-hari, baik itu pekerjaan, hiburan, maupun sesuatu yang pribadi. Tidak hanya di kota besar, kini komputer mulai menyebar hingga ke desa. Dan tidak hanya secara konvensional menggunakan Personal Computer (PC) kini juga bisa menggunakan notebook.

Selama ini dalam pengaturan hak akses, hanya mengandalkan pengaturan account dengan pemberian username dan password. Tentunya, pada masa sekarang ini hal tersebut tidaklah mencukupi, mengingat sebuah user account dapat dicuri ataupun seorang hacker dapat meretas langsung ke dalam sebuah jaringan dan langsung mengambil data-data yang diperlukan. Adanya potensi tindakan peretasan jaringan ataupun pengambilan user account dapat menyebabkans terjadinya kerawanan kerahasiaan suatu data atau dokumen sehingga diperlukan suatu aplikasi yang dapat mengamankan data, khususnya untuk shared document pada jaringan komputer lokal agar dokumen atau data tersebut hanya dapat dibaca oleh orang yang berhak [1], [2].

Kriptografi pada awalnya dijelaskan secara luas sebagai ilmu yang mempelajari bagaimana menyembunyikan pesan. Namun pada pengertian modern kriptografi adalah ilmu yang bersandarkan pada teknik matematika untuk berurusan dengan keamanan informasi seperti kerahasiaan, keutuhan data dan otentikasi entitas. Jadi pengertian kriptografi modern adalah tidak saja berurusan hanya dengan penyembunyian pesan namun lebih pada sekumpulan teknik yang menyediakan keamanan informasi. Berikut ini adalah rangkuman beberapa mekanisme yang berkembang pada kriptografi modern [3].

Algoritma RSA merupakan salah satu teknik pengamanan data dengan cara mencocokan public key yang dimiliki oleh pengirim dokumen dan penerima dokumen, yang selanjutnya dilakukan proses penguraian dengan sebuah private key. Teknik ini sangat membantu proses pengamanan file data, karena hanya orang yang punya private key saja yang dapat menguraikan isi file tersebut. [4].

Berdasarkan uraian tersebut, maka dalam proyek perangkat lunak kali ini saya akan membuat aplikasi yang dapat melakukan proses pengamanan data menggunakan algorithma RSA dan dalam teknologi informasi, telah dan sedang dikembangkan cara-cara untuk menangkal berbagai bentuk serangan semacam ini. Salah satunya yang ditempuh untuk mengatasi masalah ini ialah dengan menggunakan kriptografi dalam bentuk enkripsi dekripsi menggunakan transformasi data sehingga data yang dihasilkan tidak dapat dimengerti oleh pihak ketiga. Enkripsi mempunyai berbagai macam algoritma, salah satunya adalah algoritma Merkle Hellman Knapsack. Algoritma ini berdasarkan pada NP (Non Polynomial) complete knapsack packing problem. Ide dasar dari algoritma ini adalah memilih kejadian dari berbagai masalah yang mudah dipecahkan[5], [6], kemudian membuat penyandian yang diharapkan sulit untuk dipecahkan pihak yang tidak berwenang. Secara umum Merkle Hellman Knapsack dibuat untuk menyandikan suatu Subset Sum Problem yang disebut Super Increasing Subset Sum Problem dari modulasi multiplikasi dan permutasi.

BAHAN DAN METODE

Dalam penulisan penelitian ini, penulis melakukan beberapa hal untuk mendapatkan data yang diperlukan, antara lain [7], [8]:

1. Studi ke perpustakaan (library search)
   Dilakukan untuk mendapatkan teori-teori yang valid untuk digunakan, dikumpulkan dan dijadikan sebagai referensi tinjauan pustaka, mencari dan mengeksplorasi materi pustaka berupa buku referensi.
2. Pengumpulan data melalui surfing (field research)
   Pencarian atau penelusuran untuk mencari data yang dapat dijadikan referensi penulis yang sesuai melalui internet tentang pemahaman materi yang berhubungan dengan teknik kriptografi public RSA dan Knapsack.

2.1 Kriptografi RSA

Pada tahun 1977, Ronald L. Rivest, Adi Shamir, dan Leonard M. Adleman merumuskan algoritma praktis yang mengimplementasikan sistem kriptografi kunci publik yang disebut dengan sistem kriptografi RSA (Pahrizal dan David Pratama, 2016). Sepasang kunci yang dipakai pada kedua proses ini adalah kunci publik (e,n) sebagai kunci enkripsi dan kunci privat d sebagai kunci dekripsi dimana e, d dan n adalah bilangan bulat positif. Algoritma RSA adalah sebuah block cipher algorithm (algoritma yang bekerja per blok data) yang mengelompokkan plaintext menjadi blok-blok terlebih dahulu sebelum dilakukan enkripsi hingga menjadi ciphertext[4].

Untuk menentukan algoritma Kriptografi yang akan digunakan dalam sistem keamanan data selain pertimbangan kekuatan terhadap serangan Cryptanalis dan Bruteforce yang tidak kalah penting adalah pertimbangan kecepatan. Pada saat ini terdapat berbagai macam algoritma Kriptografi simetri maupun asimetri. Jika suatu algoritma Kriptografi dipercaya kuat namun diketahui lamba dalam proses penyandiannya maka tidak akan dijadikan pilihan oleh pengguna. Pertimbangan kecepatan ini akan menjadi lebih diutamakan lagi Ringkasan dari algoritma RSA adalah sebagai berikut :

Key Generator

• Pilih p,q p dan q prima, $p \neq q$
• Hitung n = p * q
• Hitung (n) = (p – 1)(q – 1)
• Pilih integer e ((n),e) = 1; 1 < e <
• Hitung d, d=(1+kϕ (n))/e
• Public-key KU = {e,n}
• Private-key KR = {d,n}

Enkripsi :

• Plaintext M < n
• Ciphertext C = Me (mod n)

Dekripsi

• Ciphertext C

Plaintext M = Cd (mod n)

2.2 Algoritma Knapsack

Algoritma Knapsack adalah algoritma kriptografi kunci publik yang keamanan algoritma ini terletak pada sulitnya me-mecahkan persoalan Knapsack (Knapsack Problem) (Rio Irawan Dkk, 2015). Knapsack artinya karung atau kantung.karung mempunyai kapasitas muat terbatas. Barang-barang dimasukkan ke-dalam karung hanya sampai batas kapasitas maksimum karung saja[5], [9]. Tahapan dalam membuat kunci publik dan kunci privat dalam algoritma Knapsack adalah sebagai berikut:

1. Tentukan barisan superincreasing.
2. Kalikan setiap elemen di dalam barisan tersebut dengan n modulo m.

Modulus m seharusnya angka yang lebih besar daripada jumlah semua.

1. Dekripsi dilakukan dengan meng-gunakan kunci privat.
2. Awalnya penerima pesan meng-hitung n–1 , yaitu balikan n modulo m, sedemikian sehingga n . n^1 ≡ 1 (mod m).

Kekongruenan ini dapat dihitung dengan cara yang sederhana sebagai berikut (disamping dengan cara yang sudah pernah diberikan pada Teori Bilangan Bulat):

a. n . n^1 ≡ 1 (mod m).
b. n . n^1 ≡ 1 + km
c. n–1 = (1 + km)/n, dengan k sembarang bilangan bulat

3. Kalikan setiap kriptogram dengan n–1 mod m, lalu nyatakan hasil kalinya sebagai penjumlahan elemen-elemen kunci privat untuk memperoleh plain-text dengan menggunakan algoritma pencarian menjadi superincreasing Knapsack.

Proses dekripsi dimisalkan dengan mendekripsikan ciphertext dari Contoh 4 dengan menggunakan kunci rahasia {2, 3, 6, 13, 27, 52}. Di sini, n = 31 dan m = 105. Nilai n–1 diperoleh sebagai berikut:

n–1 = (1 + 105k)/31

Dengan mencoba k = 0, 1, 2, …, maka untuk k = 18 diperoleh n–1 bilangan bulat, yaitu:

n–1 = (1 + 105 . 18)/31 = 61

Ciphertext dari proses enkripsi adalah 174, 280, 222. plaintext yang berkores-ponden diperoleh kembali sebagai berikut:

174 . 61 mod 105 = 9 = 3 + 6, berkoresponden dengan 011000

280 . 61 mod 105 = 70 = 2 + 3 + 13 + 52, berkoresponden dengan 011000

333 . 61 mod 105 = 48 = 2 + 6 + 13 + 27, berkoresponden dengan 101110

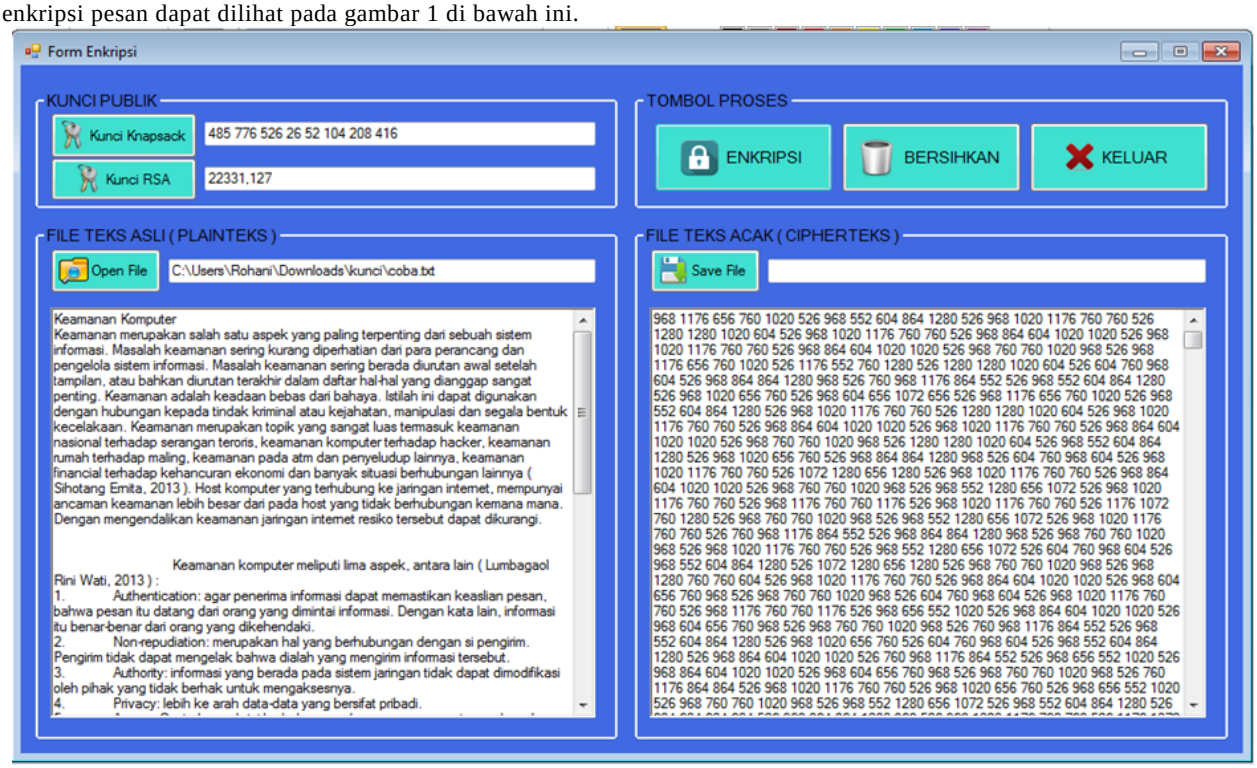
Setelah dikorespondensikan dengan kunci maka plaintext yang dihasilkan kembali adalah:

011000 011000 101110.

HASIL DAN PEMBAHASAN

3.1 Proses Enkripsi

Setelah proses pembentukan kunci berhasil dilakukan, tahap selanjutnya adalah melakukan proses enkripsi. Klik menu "*enkripsi*" yang terdapat pada form utama, lalu pilih submenu "*kunci*" kemudian setelah kunci public ditampilkan lalu klik sub menu "*open file*" setelah file tampil klik menu *"enkripsi"* sehingga akan muncul hasil dari yang kita enkripsi. Tampilan proses enkripsi pesan dapat dilihat pada gambar 1 di bawah ini.

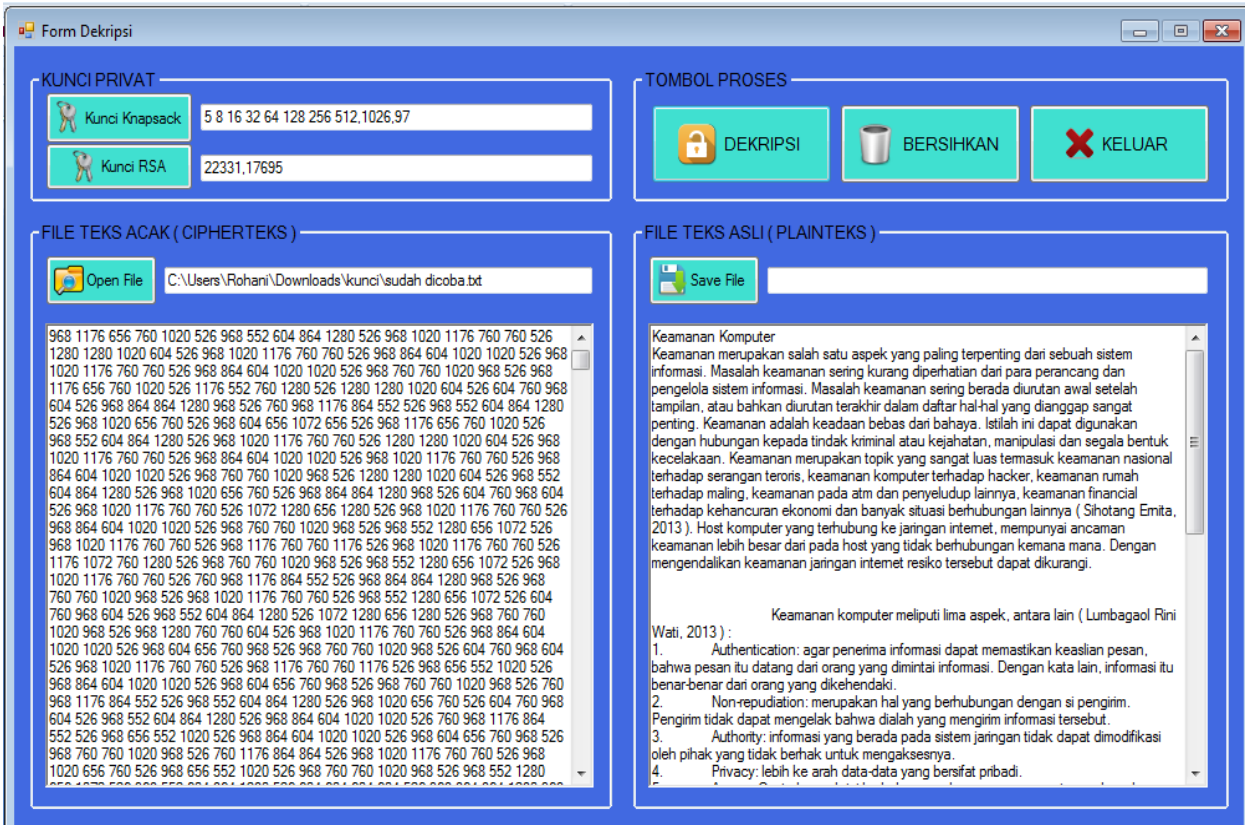


Gambar 1. Proses Enkripsi

Lalu kita pilih tombol *"Save File"* untuk menyimpan hasil tersebut. Tampilan proses save file dapat dilihat pada gambar berikut.

3.2. Proses Dekripsi

Setelah proses enkripsi berhasil dilakukan, tahap selanjutnya adalah melakukan proses dekripsi. Klik menu "*dekripsi*" yang terdapat pada form utama, lalu pilih submenu "*kunci*" kemudian setelah kunci privat ditampilkan lalu klik sub menu "*open file*" setelah itu pilih file yang disimpan tadi, lalu file akan tampil klik menu *"dekripsi"* sehingga akan muncul hasil dari yang kita enkripsi sebelumnya. Tampilan proses dekripsi pesan dapat dilihat pada gambar 2 di bawah ini.



Gambar 2. Proses Dekripsi

KESIMPULAN

Berdasarkan hasil analisis, perancangan, dan pengujian yang telah penulis lakukan maka penulis memperoleh beberapa kesimpulan, diantaranya adalah:

1. Aplikasi yang dibangun berguna untuk menyandikan data teks yang berupa file.txt yang bertujuan untuk menghindari adanya tindakan pencurian data oleh pihak yang tidak berwenang dengan menggunakan algoritma RSA dan Knapsack.
2. Semakin banyak jumlah karakter atau kalimat dalam teks maka semakin besar juga waktu yang dibutuhkan untuk proses penyandian.

DAFTAR PUSTAKA

[1] Adelia and J. Setiawan, "Implementasi Customer Relationship Management (CRM) pada Sistem Reservasi Hotel Berbasisi Website dan Desktop - Neliti," *Jurnal Sistem Informasi*, Sep. 2011. https://www.neliti.com/publications/219482/implementasi-customer-relationship-management-crm-pada-sistem-reservasi-hotel-be (accessed Apr. 17, 2021).

[2] T. Limbong, "Pengujian Kriptografi Klasik Caesar Chipper Menggunakan Matlab," *no. Sept.*, vol. 2017, 2015.

[3] R. Sadikin, *Kriptografi untuk keamanan jaringan*. Penerbit ANDI, 2012.

[4] A. Febrianto, "Enkripsi Dan Deskripsi Menggunakan Algoritma RSA," *Fak. Tek. Univ. PGRI Ronggolawe Tuban.*, vol. 53, no. 9, pp. 1689–1699, 2013, doi: 10.1017/CBO9781107415324.004.

[5] Paryati, "OPTIMASI STRATEGI ALGORITMA GREEDY UNTUK MENYELESAIKAN PERMASALAHAN KNAPSACK 0-1," *Semin. Nas. Inform.*, vol. 2009, no. semnasIF, pp. 101–110, 2009.

[6] R. Munir, "Algoritma Knapsack," pp. 0–18, 2004.

[7] T. Arianti and B. Nadeak, "Perancangan Aplikasi Pembelajaran Kriptografi Algoritma GOST dengan Menggunakan Metode Computer Based Instruction," *KAKIFIKOM (Kumpulan Artik. Karya Ilm. Fak. Ilmu Komputer)*, vol. 1, no. 1, pp. 40–46, 2019, [Online]. Available: http://ejurnal.stmik-budidarma.ac.id/index.php/jurikom/article/view/340.

[8] A. Fauzi, "Analisa Perancangan Aplikasi Penyandian Pesan Pada Email Menggunakan Algoritma Kriptografi Blowfish," *MEANS (Media Inf. Anal. dan Sist.*, vol. 1, no. 2, pp. 72–77, Dec. 2016, doi: 10.17605/JMEANS.V1I2.13.

[9] D. Rachmawati and A. Candra, "Implementasi Algoritma Greedy untuk Menyelesaikan Masalah Knapsack Problem," *J. SAINTIKOM*, vol. 12, no. 3, pp. 185–192, 2013.